

**IN THE CLAIMS:**

1 1. (Currently Amended) A method for comparing a first directory comprising unique  
2 elements with a second directory comprising unique elements, comprising:

3 (a) for each entry in the first directory, placing a hash value of the entry in a hash  
4 table, wherein the first directory is stored on a source storage system;

5 (b) selecting an entry ~~from~~of the second directory, wherein the second directory  
6 is located on a destination storage system ~~and the source storage system and the destina-~~  
7 ~~tion storage system are separate stand alone storage systems;~~

8 (c) looking up a match between a hash value of the selected entry and the hash  
9 value of the entry in the hash table;

10 (d) removing, in response to ~~locating the match between the hash value of the se-~~  
11 ~~lected entry and the hash value of the entry~~ in the hash table, the hash value of the entry  
12 ~~selected entry~~ from the hash table;

13 (e) determining if an additional second directory entry~~entries~~ exists;

14 (f) looping to step (b) in response to identifying the additional second directory  
15 entry~~entries~~; and

16 (g) reporting a difference between the first directory and the second directory in  
17 response to at least one hash value ~~first directory~~ entry remaining in the hash table.

1 2. (Currently Amended) The method of claim 1 further comprising identifying, in re-  
2 sponse to not locating the match between the hash value of the selected entry and the  
3 hash value of the entry in the hash table, that the hash value of the selected entry is sec-  
4 ond directory unique.

1 3. (Currently Amended) The method of claim 1 further comprising performing, in re-  
2 sponse to not locating the match between the hash value of the selected entry and the  
3 hash value of the entry in the hash table, a remedial function.

- 1 4. (Previously Presented) The method of claim 3 wherein the remedial function com-  
2 prises deleting the selected entry of the second directory.
- 1 5. (Currently Amended) The method of claim 1 further comprising identifying in re-  
2 sponse to no additional ~~entry~~~~entries~~ existing, any remaining hash value entry~~entries~~ in the  
3 hash table ~~data~~ as being first directory unique.
- 1 6. (Currently Amended) The method of claim 1 further comprising performing in re-  
2 sponse to no additional ~~entries~~entry existing, a remedial function.
- 1 7. (Previously Presented) The method of claim 6 wherein the remedial function com-  
2 prises deleting the selected entry of the first directory.
- 1 8. (Previously Presented) The method of claim 6 wherein the remedial function com-  
2 prises transferring the selected entry from the first directory to the second directory.
- 1 9. (Currently Amended) The method of claim 1 wherein the data are organized by a  
2 RAID system~~step of removing the selected entry from the hash table occurs in response~~  
3 ~~to identifying a match between a selected entry of the first directory and an entry of the~~  
4 ~~second directory.~~
- 1 10. (Original) The method of claim 1 wherein the hash table comprises a B-tree.
- 1 11. (Original) The method of claim 1 wherein the hash table comprises a fast lookup data  
2 structure.
- 1 12. – 16. (Cancelled)

1 17. (Currently Amended) A system for comparing a first data set with a second data set,  
2 ~~the system~~ comprising:

3 (a) means for placing a hash value of each entry of the first data set in a hash ta-  
4 ble, wherein the first data set is stored on a source storage system;

5 (b) means for selecting an entry ~~from of~~ the second data set, wherein the second  
6 data set is located on a destination storage system ~~and the source storage system and the~~  
7 ~~destination storage system are separate stand alone storage systems;~~

8 (c) means for looking up a match between a hash value of the selected entry and  
9 the hash value of the entry in the hash table;

10 (d) means for removing, in response to ~~locating the~~ match between the hash value  
11 of the selected entry and the hash value of the entry in the hash table, the hash value of  
12 the entry selected entry from the hash table;

13 (e) means for determining if an additional second data set ~~entry~~entries exists;

14 (f) means for looping to step (b) in response to identifying the additional second  
15 data set ~~entry~~entries; and

16 (g) means for reporting a difference between the first data set and the second data  
17 set in response to at least one hash value ~~first directory~~ entry remaining in the hash table.

1 18. (Original) The system of claim 17 wherein the hash table comprises a B-tree.

1 19. (Currently Amended) A computer readable medium containing executable program  
2 instructions executed by a processor, comprising:

3 (a) for each entry in a first data set, program instructions that place a hash value of  
4 ~~the each~~ entry in a hash table, wherein the first data set is stored on a source storage sys-  
5 tem;

6 (b) program instructions that select an entry ~~from of~~ the second data set, wherein  
7 the second data set is located on a destination storage system ~~and the source storage sys-~~  
8 ~~tem and the destination storage system are separate stand alone storage systems;~~

(c) program instructions that look up a match between a hash value of the selected entry and the hash value of the entry in the hash table;

(d) program instructions that remove, in response to ~~locating the match between the hash value of the selected entry and the hash value of the entry~~ in the hash table, the hash value of the entry selected entry from the hash table;

(e) program instructions that determine if an additional second data set entry ~~en-~~  
~~tries~~ exists;

(f) program instructions that loop to step (b) in response to identifying the addi-  
tional second data set entry ~~entries~~; and

(g) program instructions that report a difference between the first data set and the  
second data set in response to at least one hash value ~~first directory~~ entry remaining in the  
hash table.

20. (Currently Amended) A method for comparing a first data set with a second data set,  
comprising:

creating a hash table of hashed value entries ~~of from~~ the first data set, wherein the  
first data set is stored on a source storage system;

~~locating, for each, a hash value entry from~~ in the second data set, ~~an entry~~ in the  
hash table, wherein the second data set is located on a destination storage system ~~and the~~  
~~source storage system and the destination storage system are separate stand-alone storage~~  
~~systems~~;

removing, in response to locating ~~an the hash value entry from the second data set~~  
in the hash table, the hash value entry from the hash table ~~located entry~~; and

recording, in response to at least one hash value ~~-~~ entry remaining in the hash ta-  
ble, a difference between the first data set and the second data set.

21. (Currently Amended) A method for comparing a first data set with a second data set,  
comprising:

3           creating a hash table of hash value entries of the first data set, wherein the first  
4   data set is stored on a source storage system, and wherein the hash table comprises one or  
5   more hashed values of the first data set;

6           ~~locating~~determining, whether for each a hashed value of entry in the second data  
7   set ~~is identical to; an a hashed value~~entry in the hash table, wherein the second data set is  
8   located on a destination storage system ~~and the source storage system and the destination~~  
9   ~~storage system are separate stand-alone storage systems;~~

10          removing, in response to determining that the hashed value of the second data set  
11   is identical to the~~locating an hashed value~~entry in the hash table, the ~~located identical~~  
12   hashed value~~entry~~ from the hash table;

13          recording, in response to determining that the hashed value of the second data set  
14   is not identical to the~~locating an hashed value~~entry in the hash table, ~~that the hashed~~  
15   value~~entry~~ in of the second data set ~~is a~~ second data set unique; and

16          reporting a difference between the first data set and the second data set in re-  
17   sponse to at least one ~~first data set~~ hashed value entry remaining in the hash table.

1   22. (Currently Amended) A method for comparing a first data set with a second data set,  
2   comprising:

3           (a) selecting an entry from the first data set, ~~wherein the first data set is stored on~~  
4   ~~a source storage system;~~

5           (b) determining if a hashed value of the selected entry ~~offrom~~ the first data set is  
6   in a hash table, wherein the hash table comprises one or more hashed values of the first  
7   data set;

8           (c) adding, in response to determining that the ~~selected hashed value entry from of~~  
9   the selected entry of first data set is not in the hash table, the ~~selected hashed value~~entry  
10   ~~offrom the selected entry of the~~ first data set to the hash table;

11          (d) removing from the hash table, in response to determining that the selected  
12   hashed value entry from of the selected entry of the first data set is in the hash table, the  
13   hashed value of the selected entry from of the first data set;

(e) selecting an entry from the second data set, ~~wherein the second data set is located on a destination storage system and the source storage system and the destination storage system are separate stand-alone storage systems;~~

(f) determining if a hashed value of the selected entry ~~offrom~~ the second data set is in the hash table, ~~wherein the hash table further comprises one or more hashed entries of the second data set;~~

(g) adding, in response to determining that the hashed value of the selected entry ~~from of~~ the second data set is not in the hash table, the hashed value of the selected entry ~~of from~~ the second data set to the hash table;

(h) ~~removing from the hash table,~~ in response to determining that the hashed value of the selected entry ~~from of~~ the second data set is in the hash table, the hashed value of the selected entry ~~from of~~ the second data set ~~from the hash table;~~

(i) ~~independently continuing steps-~~(a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and

(j) reporting a difference between the first data set and the second data set in response to at least one hashed value~~entry~~ remaining in the hash table.

23. (Currently Amended) The method of claim 22 wherein the ~~step of adding the~~ hashed value of the selected entry ~~from of~~ the first data set to the hash table further comprises ~~including adding~~ information with the hashed value of the selected entry ~~from of~~ the first data set identifying the hashed value of the selected entry ~~from of~~ the first data set as originating from the first data set.

24. (Currently Amended) The method of claim 22 wherein the ~~step of adding the~~ hashed value of the selected entry ~~from of~~ the second data set to the hash table further comprises ~~including adding~~ information with the hashed value of the selected entry ~~from of~~ the second data set identifying the hashed value of the selected entry ~~from of~~ the second data set as originating from the second data set.

1 25. (Currently Amended) The method of claim 22 wherein the data are organized by a  
2 RAID system~~step of removing the selected entry from the second data set from the hash~~  
3 ~~table occurs in response to identifying a match between a selected entry from the second~~  
4 ~~data set and an entry from the first data set.~~

1 26. (Currently Amended) The method of claim 22 further comprising:  
2 (k) recording all hashed value entries remaining in the hash table as being unique  
3 to either the first data set or the second data set.

1 27. (Original) The method of claim 22 wherein the hash table comprises a B-tree.

1 28. (Original) The method of claim 22 wherein the hash table comprises a fast lookup  
2 data structure.

1 29. (Original) The method of claim 22 wherein the first data set comprises a set of direc-  
2 tory entries on a source system.

1 30. (Original) The method of claim 22 wherein the second data set comprises a set of di-  
2 rectory entries on a destination system.

1 31. (Original) The method of claim 22 wherein the first data set and second data set are  
2 on different storage devices.

1 32. (Currently Amended) A system for performing a consistency check of a source direc-  
2 tory replicated to a destination directory by comparing entries in the source and destina-  
3 tion directories, ~~the system~~ comprising:

4 one or more storage disks/devices configured to store one or more entries of a  
5 group consisting of the source directory and the destination directory; and

6 a process configured to compare entries in the source directory with entries in the  
7 destination directory by storing a hash value of each entry of the source directory and the  
8 destination directory in a hash table, the process further configured to remove from the  
9 hash table any hash value which matches any hash value of the source directory and the  
10 destination directory walking the source and destination directories only once, whereby  
11 utilization of storage subsystems associated with the source and destination directories is  
12 limited by only walking each of the source and destination directories once, and further  
13 configured to report a difference between the source directory and the destination direc-  
14 tory, wherein the source directory is located on a source storage system and the destina-  
15 tion directory is located on a destination storage system and the source storage system  
16 and the destination storage system are separate stand alone storage systems; and  
17 the process is further configured to remove matching entries from a hash table,  
18 whereby future look up operations in the hash table are enabled to be performed faster  
19 due to a smaller size of the hash table.

1 33. (Original) The system of claim 32 wherein the process executes on a computer asso-  
2 ciated with the source directory.

1 34. (Original) The system of claim 32 wherein the process executes on a computer asso-  
2 ciated with the destination directory.

1 35. (Cancelled)

1 36. (Currently Amended) A system for performing a consistency check of a source direc-  
2 tory and a destination directory by comparing entries in the source and destination direc-  
3 tories, the system comprising:  
4 one or more storage devices configured to store one or more entries of a group  
5 comprising of the source directory and the destination directory; and



6        a process configured to compare entries in the source directory with entries in the  
7        destination directory by storing a hash value of each entry of the source directory in a  
8        hash table, the process further configured to remove any entry from the hash table which  
9        matches any hash value of the destination directory~~the source directory stored on a source~~  
10       ~~storage system;~~

11       ~~the destination directory stored on a destination storage system, wherein the~~  
12       ~~source storage system and the destination storage system are separate stand-alone storage~~  
13       ~~systems; and~~

14       ~~a processor configured to select alternating entries from the source and destination~~  
15       ~~directories to be added to a hash table and further adapted to remove matching entries~~  
16       ~~from the hash table, whereby a size of the hash table is limited to a number of dissimilar~~  
17       ~~entries of the source and destination directories, and further configured to report a differ-~~  
18       ~~ence between the source directory and the destination directory in response to the number~~  
19       ~~of dissimilar entries being greater than zero.~~

1       37. (Currently Amended) A computer readable medium containing executable program  
2       instructions executed by a processor system for comparing entries in a source directory  
3       with entries on a destination directory to ensure consistency of replicated data between  
4       the source and destination directories, the system comprising:

5                (a) program instructions that select an entry from a first data set, wherein the first  
6                data set is stored on a source storage system;

7                (b) program instructions that determine if a hashed value of the selected entry of  
8                the first data set is in a hash table, wherein the hash table comprises one or more hashed  
9                values of the first data set;

10               (c) program instructions that add, in response to determining that the hashed value  
11               of the selected entry of first data set is not in the hash table, the hashed value of the se-  
12               lected entry of the first data set to the hash table;

13 (d) program instructions that remove from the hash table, in response to determin-  
14 ing that the hashed value of the selected entry of the first data set is in the hash table, the  
15 hashed value of the selected entry of the first data set;

16 (e) program instructions that select an entry from a second data set, wherein the  
17 second data set is stored on a destination storage system;

18 (f) program instructions that determine if a hashed value of the selected entry of  
19 the second data set is in the hash table, wherein the hash table further comprises one or  
20 more hashed entries of the second data set;

21 (g) program instructions that add, in response to determining that the hashed value  
22 of the selected entry of the second data set is not in the hash table, the hashed value of the  
23 selected entry of the second data set to the hash table;

24 (h) program instructions that remove from the hash table, in response to determin-  
25 ing that the hashed value of the selected entry of the second data set is in the hash table,  
26 the hashed value of the selected entry of the second data set;

27 (i) program instructions that continue (a) through (d) and (e) through (h) respec-  
28 tively for all entries in the first and the second data sets until both the first and the second  
29 data sets have been completely processed; and

30 (j) program instructions that report a difference between the first data set and the  
31 second data set in response to at least one hashed value remaining in the hash table  
32 the source directory stored on a source storage system;

33 the destination directory stored on a destination storage system, wherein the  
34 source storage system and the destination storage system are separate stand-alone storage  
35 systems; and

36 a computer associated with at least one of the source and destination directories,  
37 the computer comprising a directory comparison process configured to perform a com-  
38 parison of entries in the source and destination directories by walking each directory once  
39 and placing entries in a hash table and further configured to remove matching entries  
40 from the hash table, whereby computational cost is reduced for future look-up operations  
41 in the hash table.

1 38. (Currently Amended) The computer readable medium system of claim 37 further  
2 comprising program instructions that wherein the directory comparison process is further  
3 configured to alternate in selecting entries from the source and destination directories  
4 when walking the source and destination directories.

1 39. (Currently Amended) The method of claim 1 wherein ~~the step of~~ reporting comprises  
2 recording the difference on a disk storage device.

1 40. (Currently Amended) The method of claim 22 wherein ~~the step of~~ reporting com-  
2 prises recording the difference on a disk storage device.

1 41. (Currently Amended) The system of claim 32 wherein the process is further ~~adapted~~  
2 configured to add to the hash table any hash value which does not match any hash value  
3 of the source directory and the destination directory report the difference by recording the  
4 difference on the storage disks.